AD-A067 830 NAVAL RESEARCH LAB WASHINGTON D C F/G 7/4
ARIS MANUAL.(U)
APR 79 K HAIN
UNCLASSIFIED NRL-MR-3954 NL

END
DATE FILMED
6 -79
DDC

LEVEL II

NRL Memorandum Report 3954

ARIS Manual

KLAUS HAIN

Plasma Physics Division

April 9, 1979

This research was sponsored by the Defense Nuclear Agency under Subtasks S99QAXHC066; Work Unit Code 10, title "Late Time Electrostatic," and S99QAXHC062; Work Unit Code 41, title "Late Time MRHYDE."

NAVAL RESEARCH LABORATORY
Washington, D.C.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: (14) NRL-MR-3954
NRL Memorandum Report 3954

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
(6) ARIS MANUAL

5. TYPE OF REPORT & PERIOD COVERED
(9) Interim report on a continuing NRL problem

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
(10) Klaus Hain

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Naval Research Laboratory
Washington, D. C. 20375

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
NRL Problems 67H02-27E and 67H02-42
(Continued)

11. CONTROLLING OFFICE NAME AND ADDRESS
(11) 9
Defense Nuclear Agency
Washington, D. C. 20305

12. REPORT DATE
April 1979

13. NUMBER OF PAGES
42

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)
(12) 43 p.

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)
(16) S99QAXH, S99QAXH
(17) C0663 C0662

18. SUPPLEMENTARY NOTES
This research was sponsored by the Defense Nuclear Agency under Subtasks S99QAXHC066; Work Unit Code 10, title "Late Time Electrostatic," and S99QAXHC062; Work Unit Code 41, title "Late Time MRHYDE."

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Chemical rate equations
Differential equations generator
Automatic numerical integration
Stiff equation
FORTRAN program generator

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)
A computer system for the automated (numerical integration) of chemical type equations has been developed. The manual describes the use of the system, its input and output, and the incorporation of the generated FORTRAN program in an user program. The developed ARIS system has the following features which are essential for the successful use of such a system: 1) It is complete which means the input to the system is sufficient to generate a complete "chemistry" program (including error printouts). The generated program is efficient (vectorizable) and simple to incorporate in the user program. 2) The reaction are to be given in a physical form: A + B + . . . C + D + . . . with reaction
(Continued)

next page

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

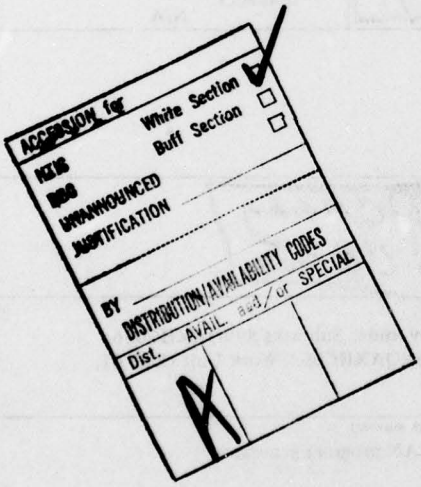
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS (Continued)

Subtask Nos. S99QAXHC062 and S99QAXHC066

20. ABSTRACT (Continued)

energies E, C, . . . The programming for initiation, back substitution, rate coefficients, conversion factors and additional source terms is easy. The generated FORTRAN is comprehensible. 3) The ARIS system provides integration schemes suited for stiff equations with special features for asymptotic behavior.



CONTENTS

ARIS MANUAL

I. INTRODUCTION

A generalized computer system for the automated (numerical) integration of chemical reaction rate equation has been developed. The system is intended to be used in conjunction with large hydrocodes therefore the integration scheme must be simple and efficient.

In order that such a system will be useful for this purpose it has to fulfill the following criteria:

1) It has to generate a complete (FORTRAN) program which contains all "chemistry", from the user input to the system. The generated program must be comprehensible (and always correct). It must also contain the possibility of an understandable error printout.

2) The input for the reaction must follow the physical notation. The further program input must be easy and complete.

3) The integration schemes provided for the numerical integration must be simple and suited for stiff equations with special features for asymptotic behavior.

The main features of the ARIS system are:

a) Input in the physical form of rate equation for particle densities and temperatures which are changed by the reactions.

b) Source terms can be included with a special feature if they represent stiff terms.

Manuscript submitted January 9, 1979.

c) FORTRAN like programming for initial conditions, rate coefficients and returning the integration results.

d) There is a choice of three second order-integrators which are suited for handling stiff equations.

e) An easy communication with the user program is made possible by allowing users common blocks (and other statements) to appear in the FORTRAN program created by the system.

f) As the ARIS system is intended to generate automatically a FORTRAN program for insertion into large hydro codes, the generated FORTRAN program is vectorizable on the ASC computer because the ARIS system uses the data in sequential order.

The manual describes the use of the system. It gives also general explanations of the content of the generated subroutines and a short description of the integration schemes and the integration logic. The titles to the different sections are self explanatory. The short example given at the end will make a new user familiar with needed input, the generated FORTRAN program and the output provided by the system.

II. INPUT SECTION

(1) General

The ARIS system will generate a FORTRAN program according to the input provided by the user. There are basically two different kinds of input: REACTION input and PROGRAM input. The latter is divided into five different sections for the different purposes explained below.

All input sections start with a first card which signifies the type of input. This card begins with a "#" in the first column. The next such card (or EOF) closes the current input. The input will be printed out as given except for this first card

#RCT Reaction input. Here the rate equations for particle densities and reaction energies, including conversion factors and source terms have to be given. Rates can be given as parameters or expressions. They may be computed either in AR$INT or AR$FUN.

#SET The content of this input section will be put into all ARIS subroutines which require user input. The user can define here his common blocks for utilization in the program which sets the initial conditions, computes rate coefficients, etc.

#INT Initial conditions. The user must here initialize the integration variables. The minimum values and allowed errors (if not to be set by default).

#FUN User input to compute variable reaction rates, reaction energies, conversion factors and source terms.

#RUN Whereas in FUN it is not allowed to change the integration variables, the variables can be recomputed after each integration step, i.e. renormalization of particle densities.

#END Is the inverse of the initialization. The user must specify where the results of the integration shall be stored for use in the calling program. Functions from certain terms can also be computed here. They may be only available here because the content of the arrays (ARIS names) will be destroyed by the integration of another point.

Any one of these inputs may be missing, but the sequence as indicated has to be kept. If there is not input to FUN or RUN these subroutines may be missing from the compiled routines.

ARIS does simple syntax checks on the reaction input and program input. It checks also for undefined variables and parameters in their order of appearance.

If there is any syntax error in the reaction input the program will stop after scanning this input with an error Code "2".

If there are syntax errors on the program input or undefined initial conditions, parameters or rate coefficients the program will stop after scanning the program input. The printout will then contain in addition to the input the differential equations. If the user is developing a program as a first step, he needs only the reaction input and can look at the printed differential equations.

(2) Definitions and Conventions

The input is basically free format, with the exception that a "C" in the first column indicates a comment and a "#" in the first column designates the various inputs.

DEFINITION:a STATEMENT is any string of characters which does not contain the character ";". The character ";" ends a statement.

DEFINITION: NAME can be any non empty character string of not more than eight characters, which must not contain the reaction delimiters = : , $ < > # ". They may contain FORTRAN delimiters + / ( ). Names have to be enclosed in < > with the exception of particle densities and reaction energies in the reaction input (see below).

DEFINITION: INTEGRATION VARIABLES are all names which appear on the left hand side of the generated differential equations. Variables are automatically defined as ARRAYS of dimension NDIM. They represent either particle densities or temperatures.

DEFINITION: PARAMETERS are all names either system or user defined. They are either scalars or arrays depending on the context in which they appear.

DEFINITION: COEFFICIENTS are all rate coefficients, reaction energies, conversion factors and source or sink terms. They can be either arrays or scalars.

DEFINITION: An EXPRESSION is any FORTRAN expression which may contain <NAMES>.

DEFINITION: DECLARATION statements: NAME: (parameter, expression); The declaration statement gives some directive to the ARIS system, i.e. PARAM: <NAMES>; used in reaction input to declare an integration variable as a parameter.

CONVENTION: An & as index in a USER name defines this variable or parameter as an array for ARIS. The use of & in INT and END assumes that the variable has at least the dimension NTOT.

(3) Reaction Input

3.1 Declarations:

All declaration statements have to proceed all other statements in the reaction input.

PARAM: <name 1, ..., name n>;

declares the variables listed as parameters which means they will be not integrated. There can be any number of PARAM declarations. The names may be repeated.

SUBST: <name> = α_1 < name 1> + ... + α_2 <name 2>;

substitutes the variable <name> in all differential equations. The coefficients α_j must be FORTRAN expressions. They cannot contain <names>.

PAIR: <name 1, name 2>;

A variable transformation to the eigenvariables is performed at integration time and the differential equations for the eigenvariables is solved. Useful if the eigenvalues are

vastly different for the linearized equations, i.e. for the sum and the difference of <name 1> and <name 2>.

3.2 Reaction Statement:

*a*1, ..., *an* :*R*1 = *b*1, ..., *bm* :*R*2 $*t*1 = *e*1, ... $*tk* = *ek*;

This corresponds to the following: species *a*1, ..., an are going to the species *b*1, ..., *bm* with the reaction rate coefficients *R*1 and reaction energies *e*1, ... *ek*. And the inverse reaction, the species *b*1, ..., *bm* going to *a*1, ..., an with reaction rate coefficients *R*2 and reaction energies $-e1$, ..., $-ek$.

Either reaction *R*1 or *R*2 can missing (but not both). Reaction energies are optional.

The species names have to be separated by a comma in the reaction rate statement (corresponding to a "+" in the usual physical rate equations). Each delimiter ends a preceeding name. The names here do not have to be enclosed in brackets.

3.3 Conversion Statement

$\sigma T : F$;

F is the conversion factor from energy to temperature

3.4 Source Statement

$\sigma V =$ Source; or $\sigma V =$ %Sink;

This corresponds to an additional term in the differential equation for the integration variable <*V*>. The second statement corresponds to subtraction and is treated as an stiff term (see discussion on INTEGRATION). The "%" holds for the entire right hand side.

3.5 General Explanations

All rate coefficients, energies, conversion factors and source or sink terms can be parameters or expressions. They may be computed in AR$INT or AR$FUN by a user's program. The ARIS system finds out if rate coefficients are constants or depend (either directly or indirectly) on the integration variable. They will accordingly computed in AR$ INT or AR$ FUN after the user input to such routines. The differential equation which result from the above defined reactions and conversions factor are:

$$D(<aj>)/DT = -RR1 + RR2 \quad j = 1, \ldots, n$$

$$D(<bj>)/DT = +RR1 - RR2 \quad j = 1, \ldots, m$$

with

$$RR1 = R1 <a1> \ldots <an>$$

$$RR2 = R2 <b1> \ldots <bm>$$

The differential equation for the temperatures are:

$$D(<tj>)/DT = fj\ (RR1 - RR2)\ ej \quad j = 1, \ldots, k$$

and for the source terms the additional terms

$D<V>/DT$ = source resp. $D<V>/DT = -$sink

are added.

ARIS counts the number of identical appearances of species. Thus for example

0,0 : $R = 02$;

will result in

$D<0>/DT = -2\ R\ <0>\ <0>$

$D<02>/DT = R\ <0>\ <0>$

The reaction statement:

$0^{\cdot},\ E = 0,\ E : <0^{\cdot}E>\ \$TE = E;$

will result in

$D<0^{\cdot}>/DT = <0^{\cdot}E>\ <0^{\cdot}>\ <E>$

$D<0>/DT = -<0^{\cdot}E>\ <0>\ <E>$

$D<TE>/DT = -\$TE<0^{\cdot}E>\ <0>\ <E>$ (conversion factor)

No equation for $<E>$ will appear but $<E>$ is counted as integration variable unless specified explicitly as a parameter in a parameter declaration.

(4) Program Input

4.1 General

The delimiters are the usual FORTRAN delimiters. The formulas are written in free form. ARIS finds out iteratively by looking at the context in which the NAMES appear whether they are scalars or arrays. It does simple syntax checks and tests if NAMES are defined in the sequential order of appearance. If NAMES are arrays the loops are automatically set. NAMES are only defined if they appear on the left hand side of a equal sign. A definition via a call to a subroutine does not define a name. There can be instances where a user wants to overwrite either the automatic definition procedure or the loop setting. There are two declaration statements which will allow him to do this.

4.2 Declarations

ARRAY: <name 1 ..., name n>;

The names so declared are arrays and are defined for the ARIS system. There can be any number of array declarations.

NOLOOP: expression;

There will be no loop for this expression. NAMES which appear here first will be defined as scalars. (They may be redefined later as arrays if they appear on the left hand side of an equation). The index & in users arrays will be translated to 1 and the index for ARIS names will also be translated to 1. If a NOLOOP declaration appears in #INT the correct indices for the appearance in #END will be put into this expression. A range which uses NP$ in AR$INT will be "1" in AR$END. This feature is designed for calls to subroutines but can also be used for equivalences in #SET.

SPLIT: Splits the subroutine AR$FUN into two parts; a) user input, b) the computation of the rates, conversion factors, etc. This is useful if the user input is very long. The following two declarations will change the integration logic. There are useful if some or all variables have reached an symptotic state.

ASYMP: (value): <list>; default: value = 1,

Asymp is invoked for items in the list. If no list is given the default is all variables. The timestep determination ordinarily is based solely on the relative difference between interpolation and extrapolation. With "ASYMP" it is based on the minimum of this and | creation/destruction - 1.|*value (if it exists). It will speed up the integration if some variables have reached an asymptotic value before others.

TOTAS: (value): <list>; default value = 0.1, default: all variables.

This will terminate the integration if for all variables in the list | creation/destruction - 1| < error* value where error is the input error for the variable. It will return the

asymptotic values. The integration time is set to the end integration time. NOTE: This will only be implemented if all specified variables have creation and destruction terms. If several ASYMP or TOTAS statements are given, the list will be added list. The implemented factor is the last value given.

PRINT: item 1,. . . item n;

The numerical values of the above list will be printed out. The current value for the specified point will be printed. The items can be either a name, or a list of names, or an ARIS FORTRAN variable i.e., ($EP (n, K), K = 1,3), <E> is a valid print list. Each print declaration gives a line of output. It will be printed at the beginning of the printout. The last four declarations can be anywhere in the program.

4.3 Position Statement

As ARIS statements are free format there is sometimes a need to bring fixed position statements into the FORTRAN program i.e. labels. This can be achieved by enclosing the statement in..,"The statement itself cannot contain any..". This statement does not end with ;

Example

"line 1

line 2"

The FORTRAN statement will start immediately after the first " the following line beginnings are kept.

4.4 Program input description

#SET

The content of this section is put into each ARIS subroutine to which is user input. It is placed immediately after the ARIS generated common. This feature is specifically designed to

allow user common block definitions <NAMES> cannot appear here, with the exception in an ARRAY declaration. An example is:

" COMMON/USER/CHEVAR (10000,4) "

#INT

The user must here define the initial conditions, gives the minimum values and error criteria. The initial values can i.e. be defined by a statement like:

<0> = CHEVAR (&, 3);

The statement will result in

DO (label) N = 1, Np$

(Label) $V (N, J, 1) = CHEVAR (N, 3)

where J is the number of the FORTRAN variable corresponding to the name <0> in the subroutine AR$INT and in the correct statements in AR$END for the subsequent points.

The minimum values as well as the maximal allowable error for the relative difference between interpolation and extrapolation which governs the timestep can be given for a list of integration variables. They can be functions of the integration variables and parameters but the lists cannot be changed. ERRORS and MIMIMA may not appear on right hand sides. If a name in the list is not an integration variable a warning will be given.

If Q represents either M or E the minima and errors are go be given in the form:

Q <name1 ..., name2> = expression;

or

Q<:FORALL> = expression;

If the lists overlap the results are unpredictable.

The defaults are:

E<:FORALL> = 1.E-4;

M<VAR> = 1.E-10 <VAR> ;

<VAR> is the initial value.

The default holds for all ERRORS and MINIMA not defined.

#FUN

Expressions for the computations for the variable rate coefficients, etc. have to given here. If i.e. the following reaction is given

N, 0+ : <*NO*+> *<*E*> = *N*+, 0 :<*N*+0> *<*E*>;

Assuming that the parameters <NO+> and <N+0> are functions of <TE> the statement:

<*NO*+> = expression (<*TE*>);

would give this coefficient. If the other one is defined by a table in the subroutine TABLE the following two statements would define it:

ARRAY: <N+0> ; NOLOOP:CALL TABLE (<N+0>,.<TE>, Np$);

#RUN

This section may be used to change the variables after one integration-step or it can be used to compute other functions of the intergration variables and parameters used elsewhere.

#END

The program which substitutes back the integration variables or functions of these. It is the inverse to #INT. The statement

CHEVAR(&,3) = <0>;

will store the integrated value of <0> back in the users array.

III. OUTPUT SECTION

There are three different outputs generated either directly by the ARIS system or during a run. The last one consists of the numerical values of the variables and the rates. This printout is according to the users needs. It may be completely suppressed.

(1) Print output

1.1 Differential Equations

As the first output the actual differential equations as generated by the input are printed, i.e., the reaction statement

O, H: 1.*E*−12 = *HO*: 1.*E*−24 *exp (−A/T);*

will result in the following

```
INITIAL:<0> = CV (&,1)
d <0>/dt = -1.E-12*<0>*<H>                RCT = 1
      + 1.E - 24 * exp (-A/T) x * <HO>    RCT = 1
INITIAL:<H> = 1.E 20
d <H>/dt = 1.E - 12 <0>*<H>               RCT = 1
      + 1.E-24*exp (-A/T)*<HO>            RCT = 1
INITIAL:<HO> = undefined
d <HO> dt = 1.E-12*<0>*<H>                RCT = 1
      -1.E - 24 * exp (-A/T)              RCT = 1
```

The initial values for <0> and <H> have been defined in #INT, whereas <HO> is undefined. ARIS will generate the next output but will not generate a FORTRAN program.

1.2 Name lists

As the next output a table is generated which gives the relations between FORTRAN

variables and ARIS names. A general list is given in the next table.

INTEGRATION VARIABLES	:	$V(&,I,J) --- ---- J = 1,2,3
PARAMETERS	:	$P(&,I), $PSC(I)
RATE COEFFICIENTS	:	$R(S,I), $RSC(I)
SOURCE TERMS, etc.	:	$C(&,I), $CSC(I)
RATES	:	$R(&,I) $RSC(I)
PRODUCTION	:	$CR(&,I,J), $CRSC(I,J) J =1,2
DESTRUCTION	:	$DS(&,I,J)----- J = 1,2
MINIMA	:	$M(&,I), $MSC(I)
ACCURACY	:	$ER(&,I), $ERSC(I)
REL. ERROR BETWEEN		
EXTRA & INTERPOLATION	:	$EP(&,I,J) J = 1,2
INDIVIDUAL TIME	:	$T(&)
INTEGRATION STEP		
OLD	:	$DO(&)
CURRENT	:	$DT(&)
INDICES LIST	:	LI$(&)

I = 1, . . . (number of specific occurrences) all these arrays will be in the ARIS common.

J = 1 is the starting value for the integration variables and the production and destruction terms. J = 2 is the extrapolated value of those variables. J = 3 is the interpolated value of the integration variable. For error terms J = 1 is the old error, j = 2 is the current error. The values for J = 1 will be replaced with the current values if the integration moves on integration step ahead.

As the points will be finishing in any order a list LI$ is kept which contains the indices of the user array.

In general all ARIS created FORTRAN variables contain a "$" in order not to conflict with users names defined in his common blocks. There are a few exceptions, the main ones being the variables N, M, L used as indices in the ARIS created FORTRAN program. These names cannot be used in the users common-blocks, as they will be destroyed.

(2) FORTRAN Program

2.1 Subroutine AR$MAI (parameters)

The system will generate a FORTRAN program. An ARIS main subroutine AR$MAI is generated which in turn will call the other ARIS subroutines. It calls according to the user provided input the different ARIS subroutines whose functions are described below. The parameters and the calling sequence of AR$MAI are described in the next chapter "SYSTEMS DESCRIPTION AND USE."

2.2 Subroutine AR$INT

Handles the initial conditions and sets up starting values. User input to this routine will consist of initialization values for species and temperatures, parameter definitions and the definition of minimum values for the variables and the maximum allowed relative errors. These statements will be repeated automatically in AR$END for the integration of new points if some of the points have been finished.

2.3 Subroutine AR$STP

Does the actual integration. No user input. There are three different integration schemes possible described briefly as a comment at the beginning of each routine. For a discussion see INTEGRATION SCHEMES. This routine handles also the integration logic which will be described later. As this program has to be vectorized, it handles the logic by multiplication and construction of functions which are either 0. or 1. This involves more computations but the data flow is uninterrupted and sequential. This routine calls AR$FUN for the computation of

the variable coefficients and AR$RHS for computation of the rates, the total production and destruction terms.

2.4 Subroutine AR$FUN

Computes the variable rate coefficients, reaction energies, conversion factors and source terms. Identical coefficients are eliminated, they will be computed only once (identical in the literal appearance). User input is the program for these computations. The system determines if rates are variable. If not, they are computed only once in AR$INT and again in AR$END if there are needed for the initialization of a new point.

2.5 Subroutine AR$RHS

Computes the rates from the rate coefficients and particle densities. The total production and destruction terms are then combined out of these rates. The source terms which may have occured in the reaction input are added. A similar procedure is used for the terms in the temperature equations. There is no user input to this routine.

2.6 Subroutine AR$RUN

Allows the user to redefine the integration variables after one integration step, i.e. renomalization of densities. If no input this routine will not be generated.

2.7 Subroutine AR$END

Handles the logic for the finishing of points. If a point is finished either in a normal or faulty manner the user programmed resubstitution of the data for this point is executed. The user specified calls to the ARIS print routine are checked and the tables printed out accordingly. The next point is initialized and put into the place of the finished one. The initialization will be repeated as long as there are points left for which the integration has not been begun. If there are no such points left the last point in the integration array will be put into the place of the

one just finished and the number of integration systems (NP$) will be decreased by one. In this manner the integration array is always densely filled. If either all points are integrated or the number of faulty returns has exceeded the user specified number control is returned to the calling program.

2.8 Subroutine AR$PRI

Prints at its first call the time the ARIS program has been created and the execution time and the values specified in print declaration (if any). Three tables can be chosen by specifying the parameter LPRI. They are described in increasing order of the index of LPRI.

1) Variables (densities and temperatures), their total production and net production (production - destruction) terms. The values are the ones at the end of the last successfull integration step (See INTEGRATION).

2) This table resembles the input for reactions. The rates and rate coefficients are replaced by their numerical values. The reaction energies do not appear. If a variable is not changed by this reaction the name does not appear. The source terms appear in a separate table which includes the temperatures. The values of the rates and the rate coefficients are those used in the integration. They are computed out of the extrapolated values (see INTEGRATION).

3) All rates and source terms will appear under the variable name with the reaction number which created them. As these rates are the same for all species involved in a reaction this will generate much more output than Table 2. The numerical values for the extrapolated and interpolated variables are given. For the temperatures only the source terms appear. The rates and the total production and destruction terms are those currently used, employing the extrapolated values. The destruction terms may differ from the sum of the destruction terms as destruction/variable rather than the destruction term itself is used by the integration routine.

IV. ARIS SYSTEM AND ITS EXECUTION

In order to use the ARIS system the user must first invoke the execution of the ARIS program, which is described below for the ASC. After a successful execution and a subsequent FORTRAN compiliation he has to call the subroutine AR$MAI (parameters) from his program. The parameters are described in the next section.

(1) ARIS Execution (JSL)

The following job control statement holds only for the Texas Instruments ASC. It will not be difficult to provide a similar procedure for any other computer. In order to execute the "ARIS" program one proceeds in the following way: First a MACRO ASSIGN Statement and then an actual EXECUTION-Statement.

```
/ MACASG SDUM,USERCAT/D77/BOO/HAINK1/TERMI/MACROX

/ ARIS (ARISDAT=INPUT,ARISOUT=FPROG,RES=PRINT,DIM=NDIM,OPT=N,
  COMMON=NAME,CHEM=C)
```

All parameters are optional and have default values.

ARISDAT =

INPUT name of input file.

DEFAULT: input cards following the / ARIS control card

ARISOUT =

FPROG name of FORTRAN program file.

DEFAULT: ARISOUT

RES =

PRINT name of print file.

DEFAULT: System print out.

DIM =

DIM=number of points to be integrated simultaneously. The storage space is approximately: Storage=NDIM (12 variables+3 number of reactions+6 +user defined arrays)

DEFAULT: 200

OPT = n

n = 1, 2, 3 invokes the different integrators. For discussion see below.

DEFAULT=1

COMMON = NAME

NAME common area for ARIS scratch storage space.

DEFAULT: Blank common

CHEM = C

Changes the first character of the names of the generated subroutines. All subroutines will start with CR$

DEFAULT = A

In order to enable the user to determine at which time the subroutines have been generated all subroutines (including AR$MAI) have as a comment the time at which they have been created. This will give an indication of the connection with the input if a complete recompilation is not done every time.

If the logic, which means the reactions as such, the number of variables and parameters including their order of appearance, has not been changed, only those routines in which changes occur have to be recompiled.

The user can also make changes directly in the generated subroutines by using some editor programs. The meaning and use of indices, the connection between ARIS names and FORTRAN variables has been explained in the output section.

(2) Program Execution (calling sequence)

In order to execute the compiled ARIS subroutines the user has to call the main subroutine AR$MAI with several parameters which are described below.

CALL AR$MAI(NTOT,TIMEA,TIMEB,DT0,DTMIN,LER,NEMAX,NER,NF,LPRI) where:

NTOT:

Total number of points to be integrated. This number can be greater or less than the number of points (NDIM) to be integrated simultaneously.

TIMEA:

Starting time for integration.

TIMEB:

End time for integration

DT0:

Guess for the first integration step. The integration scheme will adjust the integration stepsize automatically according to the desired accuracy. As the integration step is always to be assumes positive, different signs indicate the following:

DTO>O

Array input (dimension NTOT). The last integration stepsize is returned, such that for a following integration the starting integration stepsize is correct.

DTO<O

Scalar input. No return values.

CAUTION:

The user has to make sure that his time increments do not fall below the roundoff error. (DT<1.E-6 TIMEA). If necessary he has to set TIMEA = 0 and replace TIMEB by the difference adding TIMEA where absolute times are needed.

DTMIN:

The minimum timestep allowed. The ARIS program will increase NER by one for every point for which the integration step falls below DTMIN. The index of these faulty points will be put into LER(NER). The integration will be discontinued for this point.

LER:

An array (dimension NEMAX) will contain the number index faulty points, if any.

NEMAX:

The maximum number of faulty returns allowed, it is set by the user.

NER:

The number of faulty points. If NER>NEMAX the program will return to the calling program. The remaining points are neither finished nor are their return values set.

NF:

The total number of right hand side (function) calls. This will give an indication of the speed of the integration.

LPRI:

(dimension 3) controls the Printout.

LPRI(J) = 0 :

no print out

LPRI(J) = N :

for n points which finish first either regularly or faulty the selected table is printed.

LPRI(J) = -N :

up to n faulty points are printed out but as ARIS returns after NEMAX faulty points. No more than NEMAX points will be printed.

LPRI(1):

Variables, total production and net production terms. The values are those used for the extrapolation in the next integration step. They are the old values if the integration step was not completed (see INTEGRATION).

LPRI(2):

Gives in addition to LPRI(1), the rates and rate coefficients sorted according to reactions. This table will resemble the reaction input, but replaces the names with numerical values. The source terms are given in separate tables. Whereas the rates are given only for the densities, the source terms include the temperatures. The rates and source terms given are those used for the current integration step.

LPRI(3):

Prints the variables, total production and destruction terms, rates and source terms sorted according to species. The values are those used in the last integration step computed by using the extrapolated values. The total destruction terms may differ considerably from the sum of the rates if an integration step has not been completed and slighty in a normal integration step.

For the variables the extrapolated and interpolated values are given. By comparing these two values the user may have the possibility of finding which variable determines the timestep.

V. INTEGRATION

(1) General Remarks

As the system is intended to be used in conjunction with large hydrocodes the integration formulas should be simple and fast.

The numerical integration should be able to handle very stiff equations reasonably well including temperature equilibration. The program should never stop unless there are user mistakes in the coefficients which cause overflow or similar conditions.

The accuracy need not to be too high. It therefore seems appropriate to use a second order integrations scheme with predictor and corrector similar to CHEMEQ used intensively in the U. S. NAV. RES. LAB (1).

The integration logic is simple, namely: If the relative error between extrapolation and interpolation, or for the asymp option: the minima of this error and the |production/destruction - 1| is exceeds the error criterium by a factor of four the timestep will be reduced by a factor of four and the step will be repeated. If the error exceeds the given error criterium up to a factor of four, the timestep will be halved but the integration is advanced; if the error is less than 1/4 the error criterion the timestep will be increased by $\sqrt[4]{2}$. There are no iterations because it is usually more profitable to proceed with a smaller timestep than to spend time in iterations.

Because the timestep is reduced very quickly the user is advised to use as a first guess a step which is more likely too high than too low. For repeated integrations the individual timestep as returned after the integration should be used.

There are three different integration schemes provided. They differ in the handling of the stiff terms. This will be explained below. The last one (OPT=3) consists of the formula used by CHEMEQ (without asymptotics). Generally speaking all three schemes give essentially the same results in about the same time. The last OPT=3 will run a little bit faster if the equations

are not too stiff. OPT = 1 insures the convergence of the integration in all cases and should be used for very stiff equations.

(2) Integration Scheme

The set of differential equation for chemical reactions are in general of the form

$$\frac{dy}{dt} = -d \cdot y + c \tag{2.1}$$

where $d \cdot y$ is the destruction term and c the production term. Assuming d and c constant one can integrate directly and obtains

$$y(t) = y(o)\, e^{-d \cdot t} + \frac{c}{d}\,(1 - e^{-d \cdot t}) \tag{2.2}$$

As discussed before, the ARIS system uses a second order integration scheme. The different options are distinguished by how they approximate the factor

$$f = e^{-d \cdot t} \tag{2.3}$$

during a timestep. In general, one can replace this factor by any fractional polynomial so long as it represent the exponential function to at least the second order of $d \cdot t$. As the system should handle stiff equations where in one time step

$$d \cdot t \gg 1 \tag{2.4}$$

the factor should go to zero for large t. For second order accuracy of f can be represented by

$$f = \frac{1 - \alpha(d \cdot t)}{1 + (1 - \alpha)\, d \cdot t + \beta(d \cdot t)^2} \tag{2.5}$$

Then $1/f$ should correspond to the exponential function to at least the second power in $d \cdot t$. It follows that

$$\alpha + \beta = \frac{1}{2} \tag{2.6}$$

The different option are characterized by

$$\begin{array}{lll} opt = 1 & \alpha = 0 & \beta = 1/2 \\ opt = 2 & \alpha = 1/3 & \beta = 1/6 \\ opt = 3 & \alpha = 1/2 & \beta = 0 \end{array} \tag{2.7}$$

opt = 2 represents the exponential function exact to the third power, opt = 3 corresponds to CHEMEQ.

The integration performs first an extrapolation

$$y_e = \frac{y_o(1 - \alpha(d_o \cdot t)) + t\, c_o(1 + \beta\, d_o \cdot t)}{1 + (1 - \alpha)\, d_o \cdot t + \beta(d_o \cdot t)^2} \tag{2.8}$$

Then an interpolation given by

$$y_i = \frac{y_o\left(1 - \frac{t}{2} \cdot \alpha \cdot (d_o + d_e)\right) + \frac{t}{2} \cdot \left(c_o + c_e(1 + \beta\, d_e \cdot t\right)}{1 + \frac{t}{2} \cdot (1 - \alpha) \cdot (d_o + d_e) + \beta(t \cdot d_e)^2} \tag{2.9}$$

which will give for $\beta \neq 0$ for stiff equation

$$y_i(\text{asymp}) = \frac{c_e}{d_e} \tag{2.10}$$

The formula used in the different options are given as a comment in the corresponding integration routine (AR$STP).

If the declaration PAIR: <name1, name2>; was used the transformation to the corresponding eigenvectors are performed before the integrationstep backtransformation after the extrapolation and the intrapolation. The matrix elements are computed only once during at timestep.

(3) Integration Logic

The relative error $df(y)$ which determines the timestep is defined as follows:

$$df(y) = |y_e - y_i|/y.$$

or

$$df(y) = \min(|y_e - y_i|/y_{01}\ \text{factor} \times |de \times ye/c_e - 1|)$$

if ASYMP has been specified (and c_e exists). The prescription for determining the timestep is:

	$df(x) > 4$*error	:	$dt = \frac{1}{4}dt$	no advance
4*error	$\geq df(x) >$ error	:	$dt = \frac{1}{2}dt$	advance
error	$\geq df(x) > \frac{1}{4}$error	:	$dt = dt$	advance
$\frac{1}{4}$error	$\geq df(y)$	:	$dt = dt\,4\sqrt{2}dt$	advance

Three FORTRAN arrays ($FR) are constructed which are either 0 or 1 depending in what range of $df(y)$ lies. The advanced and the old variable, prediction and terms are multiplied by a combination of these $FR arrays such that the above procedure is applied. The same technique is used to determine the new timestep.

The starting of one point is achieved by setting the integration step length to zero which will automatically increase the timestep by the factor $4\sqrt{2}$. Therefore this factor is applied to the user given guess. Care is also taken to reach the exact TIMEB.

If TOTAS has been specified then if for all $|d.y/c - 1|$ <error factor the time is set to TIMEB and so the integration will be terminated.

VI EXAMPLE

1) INPUT

This example is a small portion of an ionospheric problem. It shows the basic features of the ARIS system. All neutral species are declared as parameters because one knows beforehand that they would not change appreciably in this altitude region. The common defined in set are for the whole problem so they are more complicated as needed for this example. The function LOOKTE is a table lookup for all FORTRAN variables in the common block RTLOOK.

The input is listed here in a form as returned by the ARIS system. The only difference between this table and the data cards is the appearence of the sequence numbers and the starting card (indicated in the comment).

```
 ARIS INPUT   ** REACTIONS ** 03/03/77 10:55

         C   #RCT
         C
         C    SHORT EXAMPLE
         C
RCT = 0   PARAM:<O,O2,O1D>;  PARAM:<N2D>;
         C DECLARE ALL NEUTRALS AS PARAMETERS
         C
RCT =  1  N+  ,O2  : <N+O2>*1.30  = N2D,O2+
                                    $TI=.045  ;
RCT =  2  N+  ,O2  ; <N+O2>*.64   = O  ;NO+
                                    $TI=3.32;
RCT =  3  NO+      : ADR(&,1)*<E> = N2D ,O
                                   :1.42E-12*EXP(-.4*<TI>);
RCT =  4  O2+    :ADR(&,2)*<E>  = O  ,O1D
                   $TE=1.5*(<TE>-<TI>)+1.25;
         C
         C  CONVERSION FACTOR FOR TI AND TE
         C
RCT =  5  @TI : 2./(3.*<E>); @TE : 2./(3.*<E>);
         C
         C  EQUILIBRATION BETWEEN TE AND TI
         C
RCT = 7   @TI = TEQONE(&)*<TE>; @TE = TEQONE(&)*<TI>;
RCT = 9   @TE =%TEQONE(&)*<TE>; @TI =%TEQONE(&)*<TI>;
         C
         C  END OF REACTION
         C
```

```
  ARIS INPUT   **   SET     ** 03/03/77 10:55

          C  #SET
          C
          C   COMMON FOR OUTSIDE VARIABLES.
          C
SET =  1  "       COMMON/ATMOS/ CO(151),CN2(151),CN(151)
SET =  1        . ,TN(151),VMID(151),CNP(151),COP(151)
SET =  1        . ,TMPI(151),TMPE(151)"
          C  COMMON FOR MOLECULES(IONISED)
SET =  2  "       COMMON/MOLION/CO2P(151),CNOP(151)"
          C  COMMON FOR LOOKUP TABLE
SET =  3  "       COMMON/RTLOOK/ AN(100,2),AN2D(100,2)
SET =  3        . ,AO(100,2),AO1D(100,2),AN2(100,1,)
SET =  3        . ,ANP(100,1),AOP(100,1),ADR(100,2)
SET =  3        . , AEX(100,12),TEQONE(100)"
          C  END OF COMMON

  ARIS INPUT  ** INITIAL    ** 03/03/77 10:55

          C
          C  #INT
          C   INITIALIZE VARIABLES AND NEUTRAL DENSITIES.
          C
INT =  1  <O> = CO(&);     <O2> = CO2(&);
INT =  3  <O1D> = 0. ;     <N2D> = 0. ;
INT =  5  <N+> = CNP(&);   <NO+> = CNOP(&);
INT =  7  <O2+> = CO2P(&);
          C   INITIALIZE TEMPERATURES
INT =  9  <TI> = TMPI(&);   <TE> = TMPE(&);
          C SET MINIMUM AND ERRORS
          C
          C  MINIMUM TEMPERATURES ARE SET BY DEFAULT
          C
INT = 11 M<H> = 1.E-6*(<O>+<O2>+<N2D>+<O1D>);
INT = 12 M<E> = <O2+>+<NO+>+<N+>;
INT = 13 M<O2+> = AMAX1(<O2+>,<H>);
INT = 14 M<NO+> = AMAX1(<NO+>,<H>);
INT = 15  M<N+,NO+,O2+> = AMIN1<1.E-6*<E>,<H>);
INT = 16  E<:FORALL> = 1.E-6;
          C  END OF INITIAL CONDITONS
          C

  ARIS INPUT  ** FUNCTIONS  ** 03/03/77 10:55

          C  #FUN
          C   LOOKTE AS A TABLE LOOKUP AS FUNCTION OF TE
          C   FOR ADR AND TEQONE
```

```
FUN =  1   NOLOOP: CALL LOOKTE(<TE>,NP$);
           C   COMPUTE THE OTHER RATE COEFFICENTS
           C  AND <E> FOR THE CONVERSION FACTOR
FUN =  2   <E> = <N+> + <O2+> + <NO+>;
FUN =  3    <LTI> = ALOG(<TE>);
FUN =  4   <H> = SIGN(.5,<TI>-.396);
           C  COMPUTE SIGNUM FUNCTION FOR COMPUTATION
           C  OF RATE.
           C
FUN =  5    <N+O2> =2.8E-10*(.5-<H>)
FUN =  5             4.8E-12*(.5+<H>)*EXP(.57*<LTI>);
           C   END OF FUNCTION EVALTIONS
           C

  ARIS INPUT  **   END      ** 03/03/77/10:55

           C  #END
           C   RETURN INTEGRATED VALUES
           C
END =  1    CNP(&) = <N+>;  CO2P(&) = <O2+>;
END =  3    CNOP(&) = <NO+>;
           C  END OF INPUT
           C
```

2)OUTPUT

The ARIS program will print the error message (if any) after the line in which the error has occurred. There are a few exceptions. The main exception is that the coefficients will be checked for FORTRAN syntax errors after establihing the differential equations. Therefore any error (warning) messages will appear immediately after the differential equations.

```
  ARIS OUTPUT DIFFERENTIAL EQUATIONS 03/03/77 10:55

        *** RATE EQUATIONS ***

EQU = 1  INITIAL <N+>=CNP(&)
          D(N+  )/DT = -<N>*<O2>*(<N+O2>*1.30)  RCT = 1
                       -<N+>*<O2>*(<N+O2>*.64)  RCT = 2
          ****
EQU = 2  INITIAL <O2+>=CO2P(&)
          D(O2+ )/DT = +<N+>*<O2>*(<N+O2>*1.30) RCT = 1
                       -<O2+>*(ADR(&,2)*<E>     RCT = 4
          ****

EQU = 3  INITIAL <NO+>=CNOP(&)
```

```
D(NO+ )/DT = +<N+>*<O2>*(<N+O2>*.64)  RCT = 2
             +<N2D>*<O>*(1.42E-12*EXP(-.4*<TI>)
             -<NO+>*(ADR(&,1)*<E>)    RCT = 3
 ****


     *** TEMPERATURE EQUATIONS ***

EQU = 4  INITIAL <TI> = TMPI(&)
          D(TI  )/DT = (2./(3.*<E>))*(          FAC = 0
                      +<N+>*<O2>*(<N+O2>*1.30)
                       *(.045)                  RCT = 1
                      +<N+>*<O2>*(<N+O2>*.64)
                       *(3.32))                 RCT = 2
                      +(TEQONE(&)*<TE>)         EXT = 0
                      %(TEQONE(&)*<TI>          EXT = 0
        ****
EQU = 5  INITIAL <TE> = TMPE(&)
          D(TE  )/DT = (2./(3.*<E>))*(          FAC = 0
                      +<O2+>*(ADR(&,2)*<E>)*
                       (1.5*(<TE>-<TI>)+1.25))   RCT = 4
                      +(TEQONE(&)*<TI>)         EXT = 0
                      %(TEQONE(&)*<TE>)         EXT = 0
        ****
*****  WARNING: USE OF" "&" IN VARIABLE RATE COEFFICENTS
        MAY CAUSE ERRORS
```

It follows part of the namelist in order to show the connection between ARIS names and FORTRAN variables.

```
 ARIS OUTPUT   NAMELIST                 03/03/77  10:55

          VARIABLES
 N+   $V(&,3,1)    NO+  $V(&,2,1)    O2+  $V(&,1,1)
 TE   $V(&,5,1)    TI   $V(&,4,1)

          PARAMETERS
 E    $P(&,4)      H    $P(&,5)      LTI  $P(&,6)
 N+O2 $P(&,7)      NO   &PSC(1)      N2D  $PSC(3)
 O    $P(&,1)      O+   $P(&,3)      O1D  &SC(2)
 O2   $P(&,2)

          CREATION TERMS
 N+   VOID(0)      NO+  $CR(&,2)     O2+  $CR(&,1)
 TE   $CR(&,4)     TI   $CR(&,3)
```

```
        DESTRUCTION TERMS
N+   $DS(&,1)     NO+  $DS(&,3)     02+  $DS(&,2)
TE   $DS(&,5)     TI   $DS(&,4)

        RATE COEFFICIENTS
RCT 1   $C(&,1)   FUN    VOID(0)   ---
RCT 2   &C(&,2)   FUN    VOID(0)   ----
RCT 3   $C(&,3)   FUN    $C(&,4)   FUN
RCT 4   $C(&,5)   FUN    VOID(0)   ----
```

3) FORTRAN PROGRAM

In the following a part of the created FORTRAN program is given. All loops are over n from 1 to NP$ are ommitted in order to show more clearly the created program.

```
      SUBROUTINE AR$FUN(J)
C +++ ARIS EXECUTION HAS BEEN AT: 03/03/77  10:55

      Aris common

      User common

      CALL LOKTE($V(1,5,J),NP$)


      table lookup as a function of TE

      $P(N,4)=$V(N,3,J)+$P(N,3)+$V(N,1,J)+$V(N,2,J)
      $P(N,6) =ALOG($V(N,4,J))
      $P(N,5) =SIGN(.5,$V(N,4,J)-.396)
      $P(N,7) =2.8E-10*(.5-$P(N,5))+4.8E-10*EXP(.57*P(N
     .         ,6))*(.5+$P(N,5))
```

The users program ends here. It follows the computations for the coefficients. One can note that the conversion factor $C(N,7) appears only once, as well as the source and sink terms for the temperatures $C(N,8),$C(N<9).

```
      $C(N,1) = $P(N,7)*1.30
      $C(N,2) = $P(N,7)*.64
      $C(N,3) = ADR(N,1)*$P(N,4)
      $C(N,4) = 1.42E-10*EXP(-.4*$V(N,4,J))
      $C(N,5) = ADR(N,2)*$P(N,4)
```

The coefficients for the energy changes follow now.

The two constants for the first two reactions are set in AR$INT

```
$C(N,6) = 1.5*($V(N,5,J)-$V(N,4,J))+1.25

 The conversion factor

$C(N,7) = 2./(3.*P(N,4))

 Source and sink terms

$C(N,8) = TEQONE(N)*$V(N,5,J)
$C(N,9) = TEQONE(N)*$V(N,4,J)
RETURN
END
```

In the following subroutine AR$RHS the rates and total creation and destruction terms are computed.Part of this routine is given

```
 SUBROUTINE AR$RHS(J)
C +++ ARIS EXECUTION HAS BEEN AT: 03/03/77  10:55

aris common

$R(N,1) = $C(N,1)*$V(N,3,J)*$P(N,2)
$R(N,2) = $C(N,2)*$V(N,3,J)*$P(N,2)
$R(N,3) = $C(N,3)*$V(N,2,J)
$R(N,4) = $C(N,4)*$PSC(3)*$P(N,1)
$R(N,5) = $C(N,5)*$V(N,1,J)
```

It follows the computation of the total creation and destruction terms.

```
$CR(N,1,J) = +$R(N,1)
$CR(N,2,J) = +$R(N,2)+$R(N,4)
```

The energy changes for the temperatures are regarded as creation term.(Only TE is given)

```
$CR(N,3,J) = $C(N,7)*(+$R(N,1)*$CSC(1)
            +$R(N,2)*$CSC(2))+$C(N,8)
```

The destruction terms are constructed similar to the

creation terms. They are divided by the variable itself at the end of the subroutine.

```
      $DS(N,1,J) = $DS(N,1,J)/$V(N,3,J)
      ....
      RETURN
      END
```

In order to understand the integration logic the essentials of the integration routine is given below.

```
       SUBRUTINE AR$STP(N$F)
C +++ ARIS EXECUTION HAS BEEN AT: 03/03/77/  10:55
C
      *** OPT = 1
C  YE = (Y+DT*CR*(1.+.5*DT*DS))/(1.+DT*(DS*(1.+.5*DS))
C
C  YI = (Y+.5*DT*(CR+CRE*(1.+DT*DSE))
C       /(1.+.5*DT*(DS+DSE*(1.+DT*DSE)))
C

      Aris common

      Factors for the errors are computed first

      REAL $FAC(3)/.25,1.,4./
      DO 8000 K=1,3
8000  $EF(1,K) = $ERSC(1)*$FAC(K)

       For all scalar error criteria

       for all destruction term compute

       $H(N,L,1) = $DT(N)*$DS(N,L,1)
       $H(N,L,2) = 1.+.5*$H(N,L,1)
```

Compute the extrapolated variables. Note the absence of the creation term in the first formula.(Only two variables are given).

```
       $V(N,3,2) = $V(N,3,1)
      .           /(1.+$H(N,1,1)*$H(N,1,2))
       $V(N,1,2) = ($V(N,1,1)+$DT(N)*$CR(N,1,1)*$H(N,2,2))
      .           /(1.+$H(N,2,1)*$H(N,2,2))
        ....
```

```
      compute the maxima

    $V(N,L,1) = AMAX1($V(N,L,1),$M(L))

      CALL AR$FUN(2)
      CALL AR$RHS(2)
```

the interpolated values are computed next after the computation of the right hande sides in AR$FUN and AR$rhs

```
      $DTH(N) = .5*$DT(N)
      $H(N,L,1) = 1. +$DT(N)*$DS(N,L,2)
      $V(N,3,3) = $V(N,1,1)
     .            /(1.+$DTH(N)*($DS(N,1,1)+$DS(N,1,2)
     .            *$H(N,1,1)))
      $V(N,1,3) = ($V(N,1,1)+$DTH(N)*
     .            ($CR(N,1,1)+$CR(N,1,2)*$H(N,2,1)))
     .            /(1.+$DTH(N)*($DS(N,2,1)*$DS(N,2,2)
     .            *$H(N,2,1)))
     ....

      compute maxima

      compare for all variables extrapolation and intrapolation

        $EP(N,L,2)=.5*$EP(N,L,1)+
     .          ABS($V(N,L,2)-$V(N,L,3))/$V(N,L,1)

      compute the signum functions for the relative errors
      and sum them over all variables.

       $FR(N,K) = SIGN(1.00001,$EF(1,K)-$EP(N,3,2))
       $FR(N,K) = $FR(N,K) +
     .          SIGN(1.00001,$EF(1,K)-$EP(N,1,2))
       ....
```

Then compute the total sign function.(Note that 5. is the number of variables)

```
      $FR(N,K) = .5 + SIGN(.5,$FR(N,K)-5.)
```

The following table shows the values of $FR and their connection with the timestep changes and advances.

K =	1	2	3		
$FR	1	1	1	DT=DT*2**(1/4)	advance
$FR	0	1	1	DT=DT	advance
$FR	0	0	1	DT=DT/2	advance
$FR	0	0	0	DT=DT/4	no andvance

The formula for the new advanced timestep then is the following

```
$DO(N) = $DO(N)*(((DTR*$FR(N,1)+1.)*$FR(N,2)
.        +.5*(1.-$FR(N,2)))*$FR(N,3)+.25*(1.-$FR(N,3)))
```

Where DTR = 2 **(1/4)-1

The formula for the advance of the variables, right hand sides and relative errors are

```
$FR(N,1) = 1.-$FR(N,3)
$V(N,L,1) = $V(N,L,1)*$FR(N,1)+$V(N,L,3)*$FR(N,3)
....
RETURN
END
```

4)PRINTOUT

An example of the diagnostic printout which may be desired by the user is given.This is acchieved by setting the values of LPRI in the calling sequence of AR$MAI correspondingly.

The first call will result in the following:

```
ARIS EXECUTION HAS BEEN AT 03/03/77  10:55
PROGRAM EXECUTION AT       03/03/77  10:58
```

Table 1)

	VALUE	PROD	DEST
N+	9.902E 07	0.000E 00	9.811E 05
NO+	2.274E 04	3.237E 05	3.263E 05
O2+	7.927E 04	6.574E 05	6.558E 05
TE	9.969E-02	-4.429E-04	9.504E-09
TI	1.007E 00	6.746E-03	9.597E-08

Table 2)

```
RCT=1  N+  : 6.574E 05  6.264E-10  FUN = O2+
RCT=2  N+  : 3.237E 05  3.048E-10  FUN = NO+
RCT=3  NO+ : 3.236E 05  1.423E 01  FUN
           : 0.000E 00  9.493E-13  fUN
RCT=4  O2+ : 6.558E 05  8.272E 00  FUN
```

Note that only the names appear which are variables and are involved in this reaction

```
 DEST. SOURCE TERMS
TE  9.504E-09 FUN   TI  9.597E-08 FUN

 PROD. SOURCE TERMS
TE  9.597E-08 FUN   TI  9.504E-09 FUN
```

table 3)

```
                N+
EXT   9.902E 07      INT   9.902E 07
  PROD                 DEST
                     TOT   9.811E 05
                       1   6.574E 05
                       2   3.237E 05

               NO+
EXT   2.274E 05      INT   2.274E 05
  PROD                 DEST
TOT   3.237E 05      TOT   3.236E 05
  2   3.237E 05        3   3.236E 05
  3   0.000E 00

               O2+
EXT   7.927E 04      INT   7.927E 04
  PROD                 DEST
TOT   6.574E 08      TOT   6.558E 05
  1   6.574E 05        4   6.558E 05

               TE
EXT   9.969E-02      INT   9.968E-02
  PROD                 DEST
TOT  -4.429E-04      TOT   9.504E-09
 S*   9.597E-08        S*  9.504E-09

               TI
EXT   1.007E 00      INT   1.007E 00
  PROD                 DEST
TOT   6.743E-03      TOT   9.597E-08
 S*   9.504E-09        S*  9.579E-08
```

VII. END REMARKS

The developed system will make it easier to include chemical reactions in large (hydro) codes. Its application will result in a better understanding of the reactions (and their coefficients) involved.

The resulting FORTRAN program is efficient (vectorized) and last but not least always correct if the reactions are allright. The printed out differential equations are integrated! Therefore there is no need to check this part (which is the most cumbersome) of the code.

The ARIS system has been developed with the ASC of NRL in mind but the aris program itself is easily transferable to other computers as it is written in standard FORTRAN. (some bit handling parts may have to be changed.) The ARIS system has been already successfully incorporated in several hydrocodes of NRL. It has been compared with older chemical routines (CHEMEQ). It gives the same results in approximate the same number of operations. For very stiff equations the integrationscheme (OPT=1) seems to be superior to the old CHEMEQ (essentially OPT=3).

VIII. ACKNOWLEDGMENTS

The author would like to thank TED YOUNG (NRL) for many fruitful discussions concerning the data handling and integration formula.

ELLIS HYMAN (SAI) and MARGRET MULBRANDON for advice concerning the basic arrangments for the reaction input and specially for the suggestion to put the diagnostic prints into the system. He would also like to thank GERTRUD HAIN (NRL) for writing the reaction input section and many suggestions for program features incorporated.

REFERENCES

(1) J. P. Boris and T. R. Young, A Numerical Technique for Solving Stiff Ordinary Differential Equations Associated with Reactive Flow Problems, Naval Research Laboratory, Memorandum Report 2611 (1973).